# Criterion B: Design

**Flowcharts:**

**Entity Flow Diagram**

**Diagram

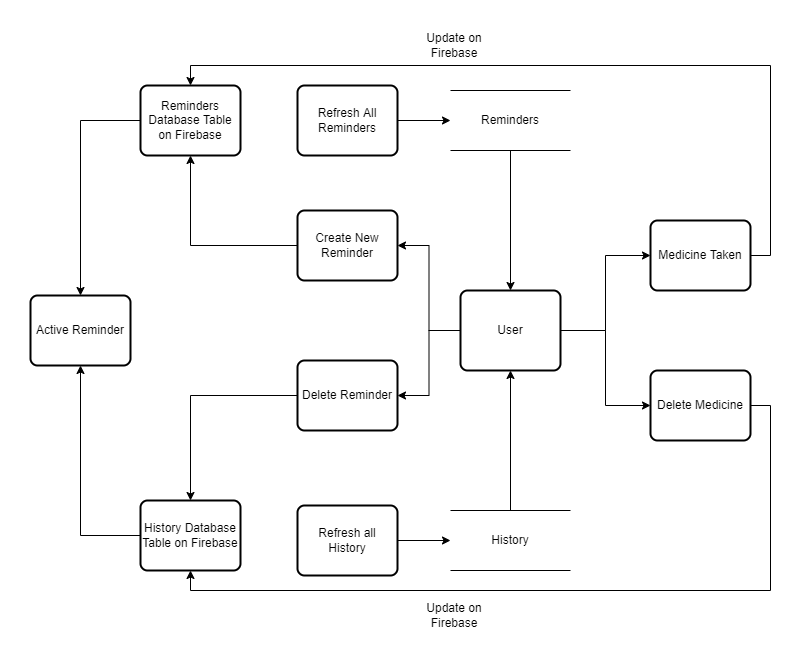
Description automatically generated**

**Hierarchical model of database**

**Diagram

Description automatically generated**

**Data Flow Diagram**

****

**System Flow Diagram**

Diagram

Description automatically generated

**User Interface Design Plannings**

The below pictures are the visualizations of the initial interface for the proposed system, which consists of 5 unique mockup screens made on *Moqups*[[1]](#footnote-1). The screenshots presented are from the *Moqups* website.

**Graphical user interface, application

Description automatically generated** Graphical user interface, application

Description automatically generated

*Welcome Screen Home Screen*

Graphical user interface, application

Description automatically generated Graphical user interface, application

Description automatically generated

*Add New Reminder Screen Home Screen after adding*

*a reminder*

**Graphical user interface, application

Description automatically generated**

*History Of Medicines Screen*

**Tables with Data type and Description for Firebase Database Storage**

**“History” Database Storage Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Other Information |
| Date Taken | Text | Date of taking medicine |  |
| Medicine name | Time | Time of Ride |  |
| No of pills | Integer | Number of pills |  |
| Time | Time | Time of taking medicine |  |

**“Reminders” Database Storage Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Other Information |
| Date | Text | Date of medicine |  |
| Medicine name | Time | Time of Ride |  |
| No of pills | Integer | Number of pills |  |
| Notes | Text | Time to take medicine | Optional Field, can be left blank by user |
| Time | Text | Provides driver with the pickup spot |  |
| User email | Text | Users email address to display the medicines for the user logged in | The user does not have to enter anything here, the app automatically enters the logged in user’s email when they add a medicine |

**“User” Database Storage Table**

|  |  |  |  |
| --- | --- | --- | --- |
| Field Name | Data Type | Description | Other Information |
| email | Text | User’s email address |  |
| name | Text | User’s name |  |
| Number | Integer | User’s phone number |  |

**Test Plan**

|  |  |  |  |
| --- | --- | --- | --- |
| Success Criteria | Test Description | Test Action | Expected Outcome |
| 1 | The user should have the option to sign up and sign into the application, once opened. | Opening the application | There will be options for the user to ‘Sign Up’ or ‘Sign-In’ and navigate to the corresponding pages. |
| 2 | When a new user signs up, his credentials should be added to the database with validation. | Keeping the password less than six characters and leaving a field blank | The username and phone number should be added to a separate database |
| 3 | When a new user signs up, the user manual should be sent to their email address | Creating a dummy user and testing | The user manual should be sent to the email address entered by the user |
| 4 | In case the user forgets their password while signing in, there should be a simple password reset process. | Using the reset function for different users | The forgot password page should send an email to the users email address with a link reset the password. Once the email is sent, a confirmation screen must be shown. |
| 5 | Upon signing in, the application must obtain the users data from Firebase | Signing in with different accounts to check if users details are retained accurately from the database | After a successful login, the application must fetch the users details and display it on a hamburger menu. Whenever a user creates a reminder, their details must be added to the document in the database |
| 6 | Easy to navigate user interface with a dashboard linking all features of the software | Testing the bottom navigation bar. | The features of the application must be accessible from the home screen. |
| 7 | The user has the ability to create a new reminder with the appropriate input. | Creating a new reminder with blank data fields. | The user will be able to create a new reminder using a form |
| 8 | Each reminder must be represented in a card with the reminder name, date and time and other required information. | Testing blank data in the cards. | A card widget will be used to display the reminders information using Google’s Material Design Guidelines. |
| 9 | If a reminder is expired, the date and time must be highlighted in red colour | Inputting a date later than the current date | The date and time of the reminder should be highlighted in red colour to show that the reminder has expired |
| 10 | Ability to refresh page to update the reminders | Update a reminder and check if it gets updated | A refresh button on the app bar of the app will reload the cards. |
| 11 | The user should be able to add the reminders to their calendar | Testing the calendar button for various reminders | A calendar entry will be created with the respective medicine details in the user’s default calendar application. |
| 12 | Once clicked on the ‘taken’ button, the reminder gets deleted from the ‘reminders’ data table and gets added to the ‘history’ data table. The ‘history’ screen can be accessed from the bottom navigation bar | Testing the taken button for different reminders | All the taken reminders will be shown in another page |
| 13 | The user should be able to email all the medicines which are taken and are present in the ‘history’ page of the app | Testing the email function | A launch url function is used to send all the data from this page to the users respective email address |
| 14 | The creator of the reminder should be able to delete the medicine reminder in the ‘history’ page. Upon clicking delete, a confirmation message is displayed in case the user accidentally clicks the button | Deleting various reminders and testing the confirmation message | The respective data is deleted from the ‘history’ data table in the database |
| 15 | The user can modify the drink water reminder interval from the hamburger menu | Testing the reminder for various values | A notification is sent to the user in the time interval selected by the user |
| 16 | The user should be able to get the location of the nearest medicine stores by clicking a button | Testing the button for multiple users | A function will be used to integrate the app with google maps |

1. "Online Mockup, Wireframe & UI Prototyping Tool · Moqups". Moqups.Com, 2021, https://moqups.com/?gclid=CjwKCAiAwKyNBhBfEiwA\_mrUMtBz976rsSLo25SRx28wx5jRh8bnZZRvzIrTgj5sCZIuJgLmEl-vzRoC2OsQAvD\_BwE. Accessed 4 Dec 2021. [↑](#footnote-ref-1)